

N72-14486

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report 32-1543

*Reliability Computation From
Reliability Block Diagrams*

Paul O. Chelson

R. Eric Eckstein

**CASE FILE
COPY**

**JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA**

December 1, 1971

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report 32-1543

*Reliability Computation From
Reliability Block Diagrams*

Paul O. Chelson

R. Eric Eckstein

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

December 1, 1971

Preface

The work described in this report was performed by the Quality Assurance and Reliability Division of the Jet Propulsion Laboratory.

Contents

I. Introduction.	1
A. Scope	1
B. Definitions	1
II. Program User's Guide.	5
A. Batch Mode	5
B. Interactive Mode	8
C. Examples	9
III. Analysis.	9
A. Notation	9
B. Theory	18
C. Summary	22
References	23

Figures

1. Reliability block diagram with active redundancy.	2
2. Reliability block diagram with standby redundancy	2
3. Block diagram that is not a reliability block diagram	3
4. Reliability block diagram with conditional probabilities	3
5. Standby redundancy in a block diagram	3
6. Reliability block diagram with dependent switching	4
7. Reliability block diagram with independent switching.	4
8. A form of standby redundancy not computable by this program	4
9. Input for example 1	10
10. Page 1 of output for example 1.	10
11. Page 2 of output for example 1.	11
12. Probability tree	11
13. Input for example 2	12
14. Page 1 of output for example 2	15
15. Output for switch option 0 for example 2	15
16. Output for switch option 1 for example 2	16
17. Output for switch option 2 for example 2	16
18. Output for switch option 3 for example 2	16

Abstract

A method and a computer program are presented to calculate probability of system success from an arbitrary reliability block diagram. The class of reliability block diagrams that can be handled include any active/standby combination of redundancy, and the computations include the effects of dormancy and switching in any standby redundancy. The mechanics of the program are based on an extension of the probability tree method of computing system probabilities.

Reliability Computation From Reliability Block Diagrams

I. Introduction

Given a reliability block diagram and the failure rates for each of the blocks in the diagram, it is often useful to calculate the reliability of the system. These calculations can become tedious except when the simplest of block diagrams is used. The computer program described in this report was developed to do these calculations for a very general class of reliability block diagrams. The input required by the program is simply the diagram, mission time, and failure rates for each of the blocks.

The program is based on an algorithm which extends the usefulness of the probability tree (see Ref. 1) to standby systems. This algorithm analytically derives the system reliability equation and then makes the computations by developing probability trees for the diagram presented rather than by using combinations of built-in equations. This method allows much more general block diagrams to be handled.

A. Scope

The program can handle reliability block diagrams of the following types:

- (1) Any active redundant system, not necessarily reducible to series-parallel. Figure 1 is an example of such a diagram.
- (2) Any standby redundant system, including dormancy and switching failure rates, with no restriction on active redundancy included in the prime or standby path. Figure 2 is an example of such a diagram.
- (3) Any combination of the above.
- (4) Partial redundant systems, where i of n blocks are needed for success.

B. Definitions

1. Reliability block diagram. A reliability block diagram (RBD) is a block diagram of a system showing all essential functions required for system operation. The purpose of the reliability block diagram is to show the relationship of essential elements to system operational success. In the simplest case, where all elements are required for successful operation of the system, all blocks are strung together in the RBD and this is called a series relationship. No redundancy exists in this type of system. When redundancy does exist, alternate modes in the system are indicated by alternate paths in the RBD. Any path through the RBD is an example of a success state for that system.

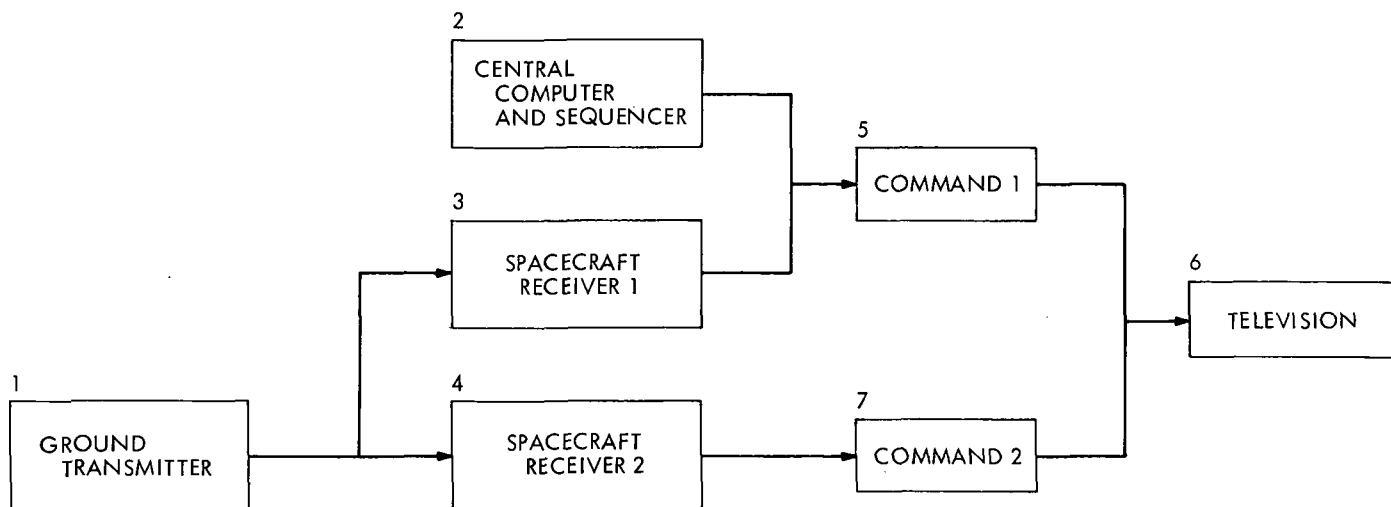


Fig. 1. Reliability block diagram with active redundancy

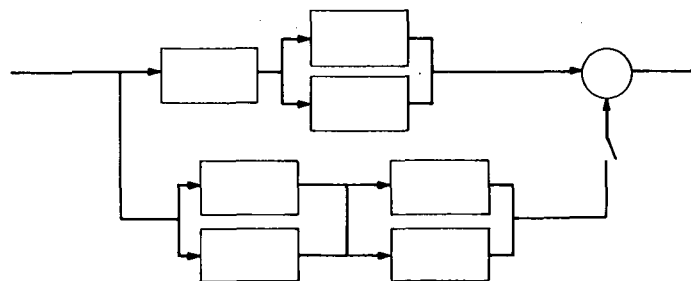


Fig. 2. Reliability block diagram with standby redundancy

The set of success states consists of all sets of blocks that can be traced from any input block, through the system, following the arrow flow (left to right, unless otherwise indicated) to an output or "success block" (a unique final block). Thus a system "success state" is a set of blocks for getting from any input to the success block. A success state does not necessarily indicate a functional flow relationship but a probability working relationship. For example, by the above definition Fig. 1 would be a reliability block diagram while Fig. 3 would not be. This is because the path through power supply 1, transmitter 1, and antenna is not a success state for this system. The frequency shifter and modulator and the receiver shown in Fig. 3 are also necessary for system success. While Fig. 3 is a functional block diagram, it is not an RBD.

It is possible to change Fig. 3 into a reliability block diagram for use in this model by the concept of conditional probabilities. This diagram (Fig. 4) is one in which we have the conditional probabilities that $P(\text{PS}_{1b} \text{ working} \mid \text{PS}_{1a} \text{ worked}) = 1$, $P(\text{PS}_{1b} \text{ working} \mid \text{PS}_{1a} \text{ did not work}) = 0$,

$P(\text{PS}_{1b} \text{ working} \mid \text{no information about PS}_{1a}) = P(\text{PS}_{1a})$, and similarly for PS_2 .¹ In other words, PS_{1b} and PS_{1a} are the same piece of equipment. Thus, if it fails as PS_{1a} , it also fails as PS_{1b} ; and if it works as PS_{1a} , it works as PS_{1b} . Thus PS_{1a} and PS_{1b} are called *equivalent blocks*.

This type of conditional probability will be called *on-on*, *off-off* conditional probability.

In cases where a functional block diagram is not an RBD, it can often be converted to an RBD by use of this equivalent block concept. The above is one example; another example can be found in Ref. 1, page 12.

A reliability block diagram with *standby redundancy* is shown in Fig. 5. (In this paper, standby redundancy is denoted by indicating a switch in the standby path and indicating the sensing and switching block with a circle.) This is a very complex RBD that by most methods is difficult to compute, especially if the switching and dormancy aspects of the standby paths are considered. In our notation, standby redundancy is denoted by using "sense blocks." In Fig. 5, blocks 47 and 49 are *sense switches* (or *sense blocks*) which control (switch in) standby blocks 41, 42, 43 and 38, 39, 40, respectively. A sense switch switches in the standby blocks it controls when the input system to the sense switch fails. Sense switch 47 would switch in standby blocks 41, 42 and 43 when block 44 failed or blocks 45 and 46 failed. Sense switch 49 would switch in standby blocks 38, 39 and 40 when block 48 failed or block 41 failed

¹The notation $P(A \mid B)$ = the probability of A given B; PS = power supply.

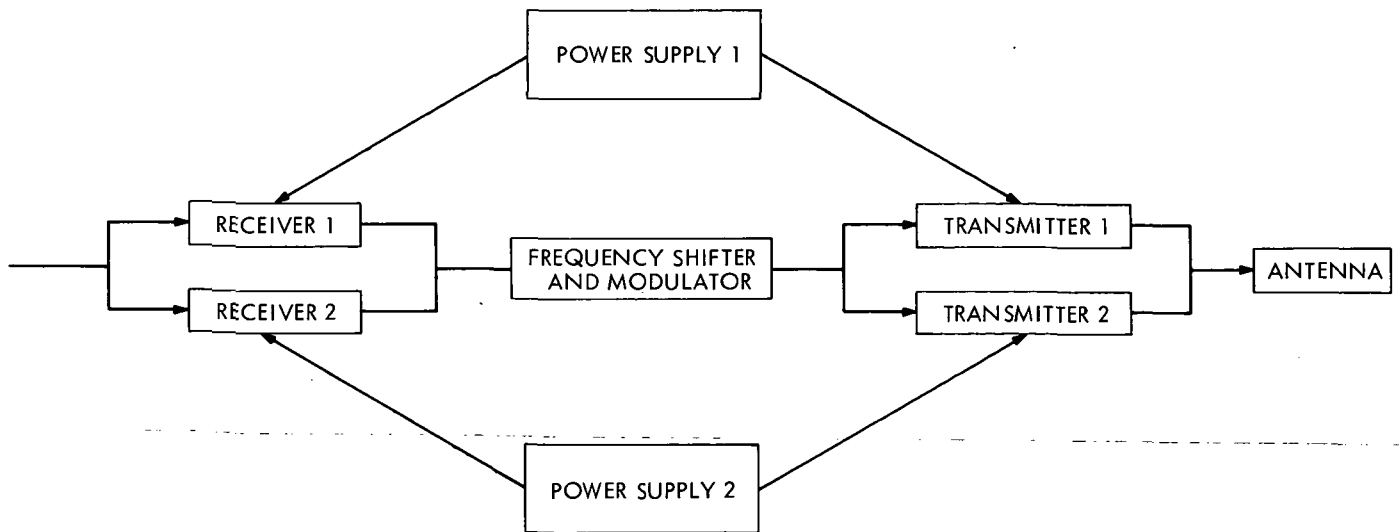


Fig. 3. Block diagram that is not a reliability block diagram

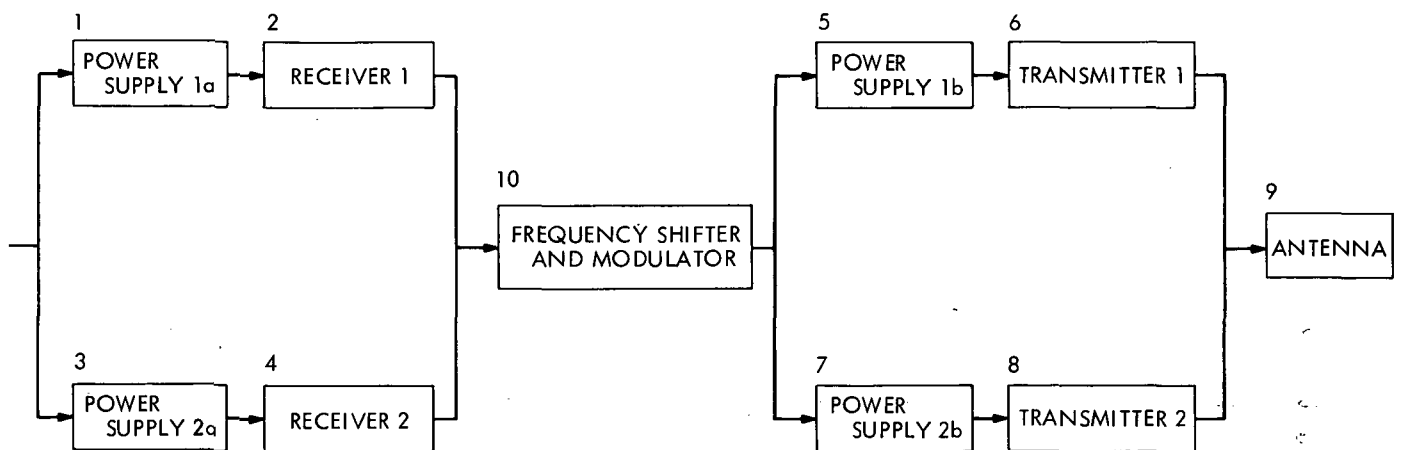


Fig. 4. Reliability block diagram with conditional probabilities

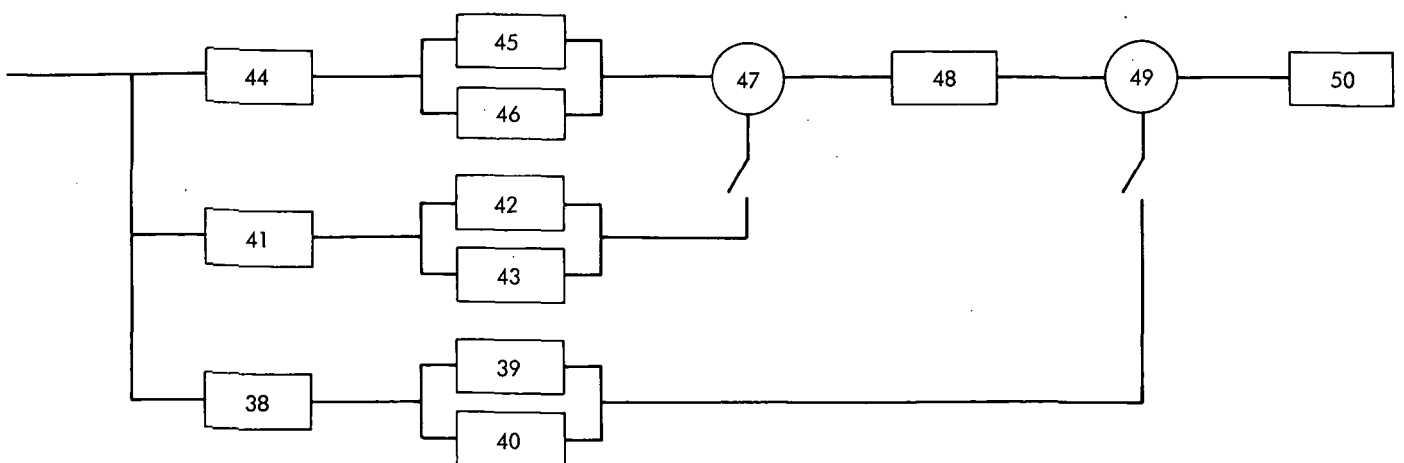
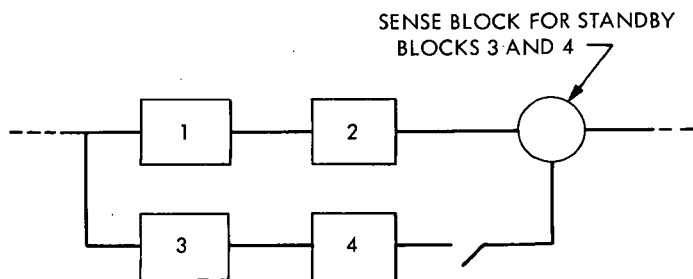


Fig. 5. Standby redundancy in a block diagram

or blocks 42 and 43 failed (the block 44, 45, 46 system would have failed previously to this last case). Thus, the sense blocks describe the actual hardware that is usually necessary to implement standby redundancy. This hardware then senses when a failure has occurred and switches in the new unit to replace the failed one if a failure occurs. It will be possible to enter failure rates and probabilities for this sensing and switching hardware, which is denoted by a sense block.

Every circuit employing standby redundancy must include some means of sensing that a failure has occurred in order to know when the standby block is needed. Analogous with this circuit operation, a convention that we require in the drawing of the RBD is that sense blocks be put into the main or prime path and that these sense blocks be specified as "controlling" their standby blocks. Thus sense blocks (or sense switches) are used here to describe standby redundancy. For example:



Note that there are some restrictions on the use of standby redundancy in this program. When several standby redundant units are drawn in a reliability block diagram, they must be nested in such a way that sense switches do not control other sense switches. The diagrams shown (Figs. 6 and 7) are equivalent only in the case of perfect switching. Figure 6 illustrates a dependent type of switching (i.e., if sense switch 35 fails, the system fails). Figure 7 illustrates an independent type of switching (if sense switch 35 fails, sense switch 45 will switch in standby blocks 4 and 5, and the system will work). The program will handle independent switching, with four switching options: 0 = perfect switching (probability of switch working = 1.0); 1 = constant probability that switch works; 2 = dormant failure rate only for switch; 3 = dormant and active failure rate for switch. Reliability block diagrams such as that shown in Fig. 7 can be computed with this program.

Unfortunately, it is sometimes not possible to redraw an RBD such that it can be computed with just a different switching assumption. For example, consider Fig. 8. It is not possible to redraw Fig. 8 so that sense switch 45 is not

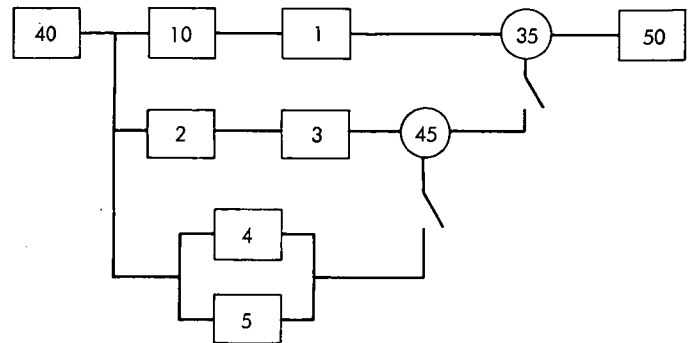


Fig. 6. Reliability block diagram with dependent switching

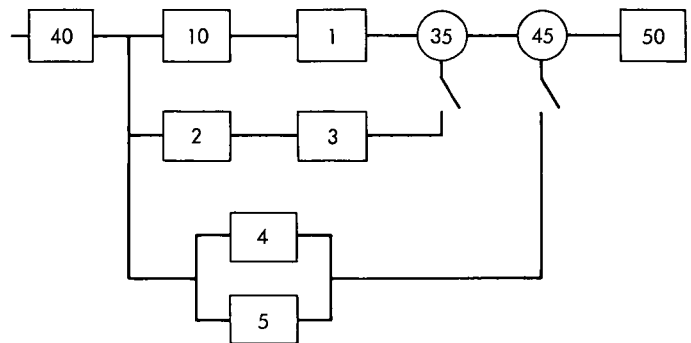


Fig. 7. Reliability block diagram with independent switching

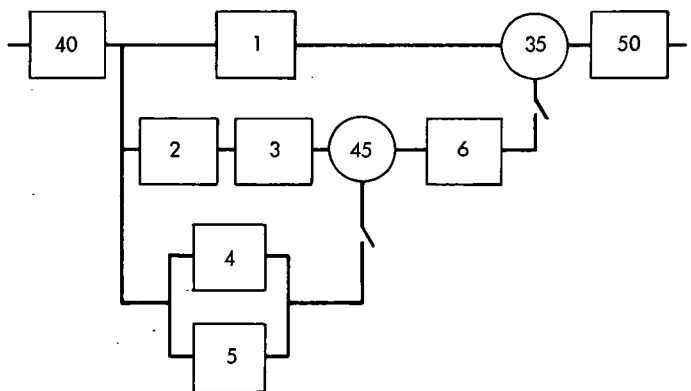
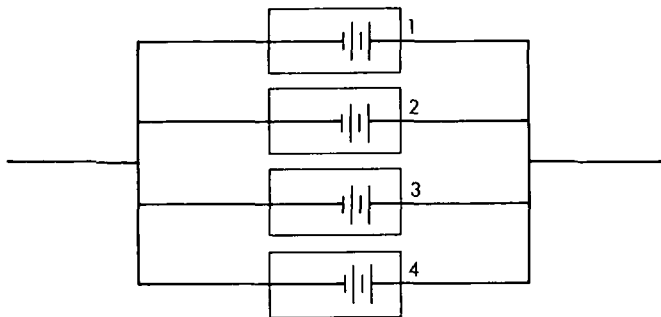


Fig. 8. A form of standby redundancy not computable by this program

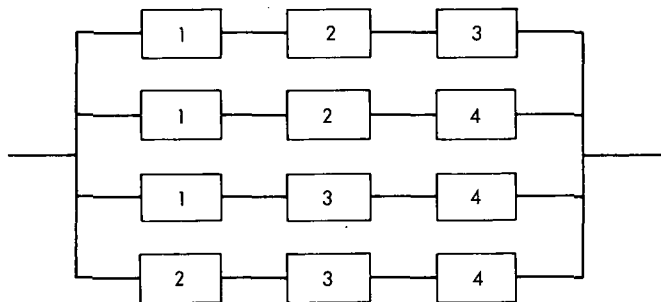
"controlled" by sense switch 35. As such, this diagram cannot be computed with the current version of this program.

Another type of redundancy that we would like to describe with RBDs is *partial redundancy* (sometimes called cooperative redundancy). This is a configuration only a portion of which need work for system success. For

example, consider four batteries connected functionally in parallel as follows:



Assume that the power requirements of the system require at least three of the four batteries to be operable. This is partial redundancy. (Note that if only one of the four batteries is needed, then the above functional diagram is also the RBD.) The above is not an RBD for this circuit since it does not describe the 3 of 4 portion of the redundancy. The above functional diagram can be redrawn as an RBD as follows:



This diagram describes the success states for the circuit. Since any of the four success states is sufficient, the four are drawn in parallel. The added constraint needed is the notation that where 1, 2, 3, and 4 appear more than once, these are "equivalent blocks," i.e., the same piece of hardware appearing more than once in the RBD. These equivalent blocks are denoted by using the same block number in the drawing. (This RBD, with the equivalent blocks specified, cannot be directly entered for computation by the program without some modification. The blocks must be numbered so that they all have different numbers; then the blocks that are equivalent are declared "equivalent blocks.")

2. **Supporting terminology.** Several other items of nomenclature are defined as follows:

- (1) Initial reliability R_{i0} is defined as the turn-on probability, that is, the probability that unit i works at time = 0. This can be set to 1.0 if one wishes to use just the exponential model.

- (2) λ and λ_d represent the active and dormant failure rates respectively of a unit in the block diagram.
- (3) Where standby redundancy is used, P_s , λ_s , λ_{sd} denote the probability that the sense and switch mechanisms work. λ_s is the failure rate for the sense and switch mechanisms when the standby path is active; λ_{sd} is the failure rate for the sensing and switching when the standby path is dormant.
- (4) In an RBD, any block without inputs is an *input block*.
- (5) A *probability tree* is a listing of the set of success paths for the system described by the RBD and is best understood by considering the examples in Section III. This term is not needed by the casual user of the program, but only by one who desires to understand the methods used in the program.

II. Program User's Guide

The program is written to be used on a UNIVAC 1108 time-sharing system with 65K core storage and a UNIVAC 1108 FORTRAN V compiler. The program has variable dimensions which allow maximum efficiency in using all of the 65K core storage. The program can be run in either batch or interactive (from a terminal) mode. Also available is a version of the program utilizing the FORTRAN V FLD statement to pack four numbers into a single word, thus freeing an additional 30K of core storage. This last version is only written for use in the batch mode.

A. Batch Mode

1. **Input.** To enter a reliability block diagram, arbitrarily number the blocks 1-50. Each block must have a unique block number. With reference to the input section, the following should be noted: Block numbers are always right-justified I2 format. Data card column numbers are shown in brackets. Any series of block numbers starts at X, the left-most column in [X-Y], and fills in successively to the right to column Y. There is one data card per block number unless otherwise specified.

a. *Diagram inputs and outputs.* This first section contains one data card for each block number in the diagram, with a maximum of 14 inputs and 14 outputs to/from each block number. The card format is arranged as follows: Block number [1-2], input to that block number [3-32], output from that block number [33-62]. The unique success block number (with no output blocks) has a 9 in [80] and is the last card in this series.

b. Standby. This section contains one data card for each sense switch in the diagram and all the standby blocks controlled (switched in) by the sense switch. There is a maximum of 15 sense switches and 29 standby blocks. The first data card is the number of sense switches. (The number of sense switches is specified in columns [1-2] in right-justified I2 format, 0 = none.) If there are sense switches, subsequent cards have sense switch number [1-2] and standby blocks controlled by that sense switch [3-60] (format 29I2).

c. Equivalent blocks. This section contains one data card for each set of equivalent blocks in the diagram. There is a maximum of 20 equivalent block numbers in a single set and a maximum of 20 sets. If equivalent blocks are in standby, every equivalent block in a set must be controlled by the same sense switch. The first data card is the number of equivalent block sets in the diagram. (The number of equivalent block sets is specified in columns [1-2] in right-justified I2 format, 0 = none.) If there are equivalent blocks, there will be one card for each set with the equivalent block numbers [1-40] for each set in format 20I2.

d. Mission time. This card contains the distribution type (1 = the exponential distribution, which is the option currently available) and mission time (in hours). The card is as follows: a 1 in [2], mission time [3-14] (E12.7 format). For example, exponential distribution with mission time = 100,000 h would have the following in columns [1-14] (b = blank in column 1):

b1 + .1000000 + 06

Note that the exponential restriction is imposed simply because other distributions have not been included in the computation phases of the program. This is not an inherent limitation of the program; any user can add equations for his own distribution at the points in the MAIN program noted by comment cards.

e. Active parameters (failure rates and probabilities). This section contains one data card for each block number in the diagram (other than sense switches) together with its R_0 and failure rate (λ). The probability R_0 is the constant probability that the block will initially turn on. If [15-24] are left blank, R_0 is set equal to 1.0 by the program. The card format is: block number [1-2], λ [3-14] (E12.7 format), R_0 [15-24] (F10.7 format). The data card for the last block number in this series has a 5 in [80].

f. Dormant parameters (failure rates or dormancy factors). This section contains provisions for assigning a

dormant failure rate (λ dormant) to each standby block in the diagram. There are three options:

- (1) No dormancy involved, assume hardware is perfect in the dormant state, insert a blank card.
- (2) Read in a dormancy factor, — blank [1-2], dormancy factor [3-14] (E12.7 format) — which the program will multiply by active λ of each standby block to yield the λ dormant for each standby block.
- (3) Read the λ dormant for each standby block individually. To do this, the first card will have a 99 in [1-2], blank [3-80]. Then there will follow one data card for each standby block, with standby block [1-2] and λ dormant [3-14] (E12.7 format). With this third option, the data card for the last standby block in the series has a 6 in [80].

g. Switching options. This section contains one data card for each sense switch designating one of the four switching options (0, 1, 2, 3).

0 = Perfect switching (probability of switch working = 1.0). Sense switch [1-2], blank [3-79], 0 [80].

1 = Constant probability that switch works. Sense switch [1-2], blank [3-14], probability [15-24] (F10.7 format), 1 [80].

2 = Dormant failure rate only for switch. This means the failure rate for the switch when its associated standby blocks are in the dormant state. Sense switch [1-2], λ dormant [3-14] (E12.7 format), 2 [80].

3 = Dormant and active failure rate for switch (2 cards per switch). The dormant and active failure rates for the switch are for the time periods when the associated standby blocks are in the dormant and active state, respectively. Sense switch [1-2], λ dormant [3-14] (E12.7 format), 3 in [80]. Sense switch [1-2], λ active [3-14] (E12.7 format), 3 in [80].

If you do not have standby redundancy, no switch or dormant cards are needed.

h. Recalculate option card. This card is blank [1-79] and has last [80] set equal to 7, 8, or 9, as follows:

- 7 = Recalculate the diagram with the new parameters. The program will loop back and start reading from Section II-A-d (mission time) through this section. This permits varying mission time, R_0 , active/dormant λ s, and switch options.

8 = Read in new reliability block diagram and parameters. The program will loop back and start reading from Section II-A-a (*diagram inputs and outputs*). This permits varying diagram configurations.

9 = End of computer run.

i. *Print options.* Various users require different amounts of information included in the printout. This was taken into account by using a printing variable called IPRINT. When not specified by the user, IPRINT is set to 0 by the program. The output when IPRINT = 0 is that which is normally needed by the user; it includes the RBD description, the failure rates and other parameters used, and the result – the value for the system reliability.

It is possible for the user to specify other values of IPRINT if desired. The options available are:

IPRINT = 0: RBD, parameters, and results are printed.

IPRINT = 2: The above plus the overall system probability trees are printed.

IPRINT = 3: The above plus all probability trees used in the computations are printed.

IPRINT = 4: The above plus diagnostic information and all \bar{R} 's and P_i 's as a function of time are printed.

The last option (IPRINT = 4) is intended for use only as a diagnostic tool. To follow the output requires following the program listing in considerable detail.

Options 0 and 2 are the options most used. The IPRINT = 0 option is set by the program; nothing need be done. If, however, the user wishes to override this, the very first card of the data deck, immediately following the @XQT card, should have IPRINT=2 in the first 8 columns of the card (or 3 or 4 for those options).

In addition to IPRINT, there is another variable which is normally set by the program but which can be overridden by the user if desired. This option is controlled by the program variable NSIG and refers to the number of "significant digits" that will be printed for the computed system reliabilities. The term "significant digits" is defined in a very special way. The "significant digits" are the non-nine digits in the reliability number. Thus .99986, .975, and .52 all have two "significant digits." The value of NSIG is set equal to 3 by the program, and therefore three "significant digits" will be printed for the system reliability unless NSIG is specified as something else by the user. To accomplish this, columns 9–15 of the first card of the data

deck (card that contains the IPRINT specification if IPRINT is also being specified) contain: ,NSIG=n, where n = 1, 2, . . . , 8. Thus if both IPRINT and NSIG are being specified by the user, the first card of the data deck should look like the following, in columns 1–15: IPRINT=2, NSIG=4

2. Output. The computer output comes in the following form:

- (1) Page 1: This page lists each block number of the reliability block diagram with its inputs and outputs. It lists each sense switch with the standby blocks it controls and each set of equivalent blocks. Check page 1 to make sure that the diagram entered is the one that was intended. (Failure to do this leads to the most commonly made error.)
- (2) Page 2: This page lists the success paths of the probability tree. It is printed only if IPRINT \geq 2. Plus numbers indicate a success; minus numbers indicate a failure. Each path of the probability tree is indicated by up to two lines of print, consisting of up to 50 numbers. Note that this information about the probability tree is not needed by the casual user. It is from this tree that one can derive the system reliability equation used in the computation phase of the program. This is also true for pages 3 and 4 described below.
- (3) Page 3: This page is printed if IPRINT \geq 2 and if the reliability block diagram has at least one sense block which controls two or more standby blocks. New pseudoblocks (denoted by 51–65) replace those blocks and the standby trees (success routes of the standby system) replaced by these pseudoblocks are printed.
- (4) Page 4: This page is printed if IPRINT \geq 2. If page 3 is printed, page 4 is printed also. Page 4 lists the original probability tree (page 2) with pseudoblocks replacing their respective standby trees. This replacement usually causes a reduction of the number of original success paths.
- (5) Page 5: This page lists, for each original block, its active and dormant failure rate, R_0 unless $R_0 = 1.0$, and its reliability unless it is a standby block replaced by a block 51–65. In the latter case, the reliability of the block 51–65 will be printed. Each sense switch will have its active and dormant failure rate and probability printed. Then the mission time and the reliability of the system are printed.

- (6) Additional pages: Not mentioned are those additional pages printed when $IPRINT \geq 3$. If $IPRINT = 3$, considerably more pages of intermediate probability trees are printed. If $IPRINT = 4$, additional arrays are printed at various points in the program to provide diagnostic information. This additional information is difficult to understand without considerable knowledge of the program and so is omitted here.

3. Error checks. There are several error checks built into the program to protect the user from added cost and erroneous results. The checks are used to detect and locate input errors in the diagram inputs and outputs section of the data deck. If an error is detected, the appropriate error message is printed, and the program advances to the next set of data. An input error may result in a valid reliability block diagram, but not the one intended. The error checks cannot detect this, so the user should compare all the block numbers and their inputs and outputs on page 1 of the printout to the diagram he intended to input. The user should check page 5 of the printout to be sure the parameters are correct.

Each error message contains the following:

- (1) The location of the error in the program.
- (2) The type of error that occurred.
- (3) The corrective measures that should be taken.
- (4) The input variables that were associated with the error.

Most errors that result from format errors in the input deck will be self-explanatory when encountered. There are four error messages that need further explanation, however:

- (1) **ERROR 902 XX.** XX is the value of JS, which is the program variable that stores the number of success paths in the probability tree. If JS is 200 or larger, there are too many success paths for the program to handle the diagram as it was entered. The diagram should be broken up and loaded as several portions.
- (2) **ERROR 613 XX.** XX is again the value of JS. ERROR 613 indicates that one of the inside entries of the probability tree is blank. This indicates an inconsistency in the block diagram as entered.

- (3) **ERROR 385 XX.** XX is the value of JS, which when displayed as ERROR 385 indicates that the DO 385 loop has been completed. As this should never happen, an error statement is printed.
- (4) **ERROR 387 XX.** XX is the value of the program variable ISA, which is the block number now being put into the tree. This error indicates that ISA is not an element of the IDR array, yet all its inputs are elements. This is contradictory since this is one of the criteria for an element of IDR. (IDR is the array which holds those blocks made inactive by the failed blocks in the path being calculated.)

The last three errors generally result when there is an ambiguity or inconsistency in the description of the block diagram inputs and outputs. Error corrections for ERRORS 613, 385, and 387 are as follows: Check the input of the blocks of the block diagram to be sure that the input/output lists correctly describe the diagram. If they do, make sure that your diagram is a reliability block diagram as defined herein.

4. General program limitations. To accommodate the storage capacity of the UNIVAC 1108, the following limitations are necessary:

- (1) The block diagrams can have at most 50 blocks. This is not a serious restriction. If the diagram is larger, it is only necessary to break it into portions, compute the portions, then enter each portion as one block on another run.
- (2) There can be at most 200 success paths. Since it is hard to know this ahead of time, an error message is printed if more than 200 paths exist. If this should happen, it is simple to break the diagram into smaller portions and proceed as in (1).
- (3) Each block can have a maximum of 14 inputs and 14 outputs.
- (4) There must be only one output block. Again, this is not serious, for if there is more than one possible output block, it is possible to process these blocks through one final success block and give this block a probability of success of 1.0 (or a failure rate of zero).

B. Interactive Mode

1. Input. The input data necessary for the interactive version of the program is identical to that of the batch mode and is also required in the same order. For user

convenience when working from a terminal, the format is slightly different. But this is self-explanatory as there is considerable interaction on the part of the program.

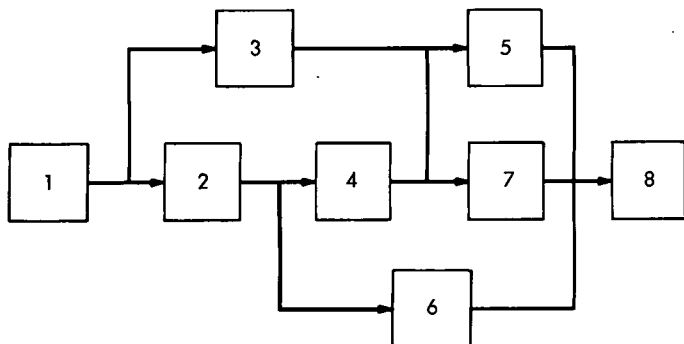
2. Output. The output is identical to the batch mode except that probability trees will not be printed. This is because probability trees are, in general, quite long and therefore undesirable to have printed on a very slow device such as a teletype. The error messages and the program limitations are the same as for the batch mode.

C. Examples

Several examples were chosen to illustrate the input/output requirements of the program as well as the capability of the method. All inputs and outputs are shown in the batch mode version of the program; this is, in general, the version used most. Since this program was designed specifically for very complex redundancy schemes, the input/output is generally larger than is conveniently handled interactively (from a terminal).

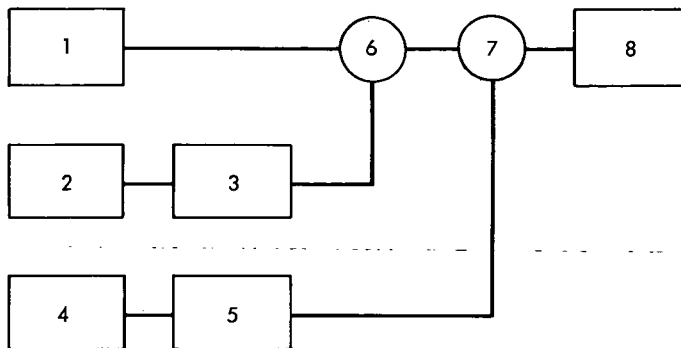
It should be stressed that the power of the program is used and needed for complex block diagrams. If one is computing very simple and straightforward RBDs, this method will often not be the most efficient. However, for complex RBDs, this method is more efficient than any other available.

1. Example 1. Consider the following reliability block diagram. Block 1 is the input; block 8 is the output. All redundancy is active. There is no standby redundancy.



This diagram was entered on the computer, failure rates of 0.0010536 being assigned to each of the blocks for a mission time of 100 hours. Figures 9-12 show the input cards required and the two pages of output if the IPRINT = 0 option is used. Also shown is the probability tree that is printed if the IPRINT = 2 option is used.

2. Example 2. This example uses standby redundancy. Blocks 2 and 3 are standby and back up the prime unit 1. Units 4 and 5 are in standby and are used if the system formed by blocks 1, 2, and 3 fails. Blocks 6 and 7 are the sense blocks that control blocks 2, 3, and 4, 5, respectively.



Failure rates were assumed as shown below in the list of input cards. A dormancy factor of 0.01 was used for the units in standby. The recalculate option number 7 was used, and the switch option was varied. Thus the output consists of the block diagram plus four pages of results for each of the switch options (Figs. 13-18). The IPRINT = 0 option was used, and therefore there are no probability trees in the output. Note that blocks 51 and 52 appear. These are pseudoblocks set up for blocks 2, 3, and 4, 5, respectively. The value shown under reliability for these pseudoblocks is $\bar{R}(t = \text{mission time})$. This information is not of interest to the casual user.

3. Example 3. The RBD below (Fig. 19) is one that includes considerable standby redundancy. Blocks 8, 9, 3, 11, 12, 17, 18 are all standby redundant units that are dormant as long as their associated prime paths are working. Blocks 2, 6, 14, 16 are the sense switches for controlling the standby redundancy.

The input sheets, which should now be clear, are not shown. However, Fig. 20 presents the output obtained when this diagram is run. One can see below, in the output listing, the failure rates that were assumed. (IPRINT option 0 was used; NSIG = 7 was used.)

III. Analysis

A. Notation

The following notation and terms must be developed before the analysis of the standby case is discussed:

- (1) When a standby path consists of more than one block, the blocks in the standby path are lumped

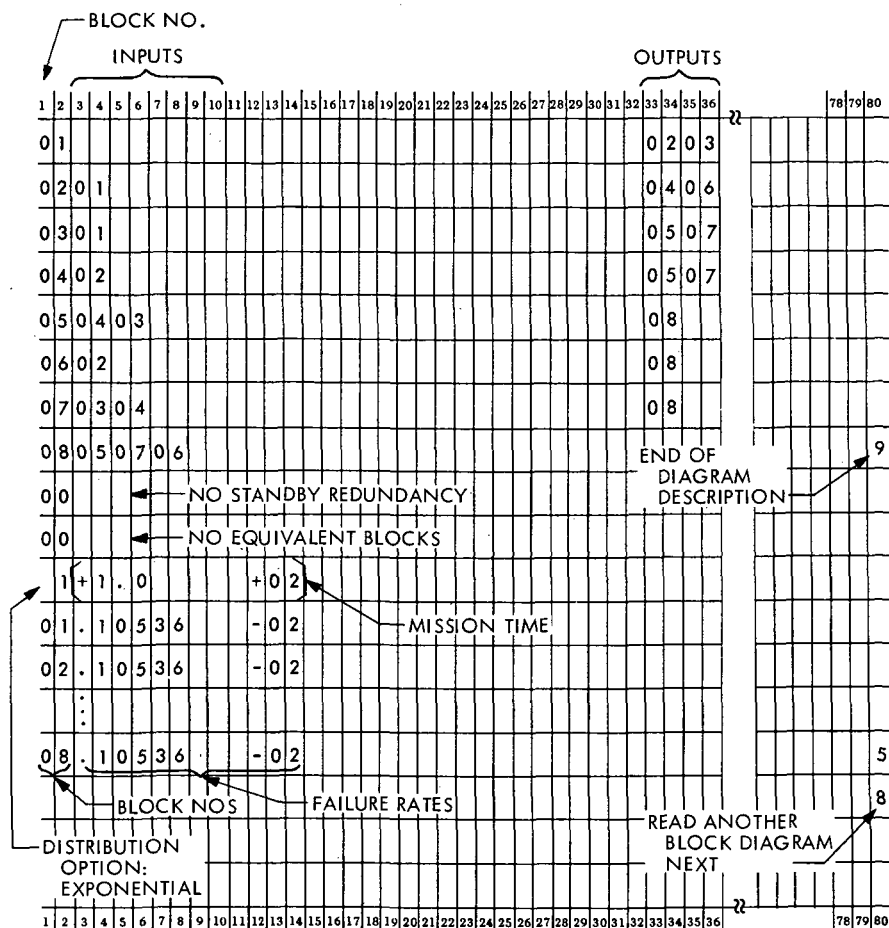


Fig. 9. Input for example 1

together and called a *pseudoblock* for portions of the analysis. (The user need not concern himself with pseudoblock. The program sets them up internally when needed. However, to understand the analysis that the program is based on, it is useful to introduce them here.)

- (2) $\bar{R}_i(t)$ is the probability that block i (or pseudoblock i) works, given that it is needed.
- (3) R_{i0} is the turn-on probability of block i , i.e., the initial reliability of block i at time = 0.
- (4) $R_i(t)$ is the reliability of blocks i through time t .
- (5) $P_i(t)$ is the probability that block i (or pseudoblock i) is needed. In other words, P_i is the failure probability of the prime path, or sub-RBD, for which block i is the standby.
- (6) $\dot{P}_i(t)$ is the first derivative of P_i .

RELIABILITY BLOCK DIAGRAM				
BLOCK 1	INPUT	OUTPUT 2	3	
BLOCK 2	INPUT 1	OUTPUT 4	6	
BLOCK 3	INPUT 1	OUTPUT 5	7	
BLOCK 4	INPUT 2	OUTPUT 5	7	
BLOCK 5	INPUT 4	OUTPUT 9	3	
BLOCK 6	INPUT 2	OUTPUT 9		
BLOCK 7	INPUT 3	OUTPUT 9	4	
BLOCK 8	INPUT 5	OUTPUT 7	6	

Fig. 10. Page 1 of output for example 1

	ACTIVE F/R	DORMANT F/R	R-INITIAL	RELIABILITY
BLOCK 1	.1053605-02	.0000000		.9000000+000
BLOCK 2	.1053605-02	.0000000		.9000000+000
BLOCK 3	.1053605-02	.0000000		.9000000+000
BLOCK 4	.1053605-02	.0000000		.9000000+000
BLOCK 5	.1053605-02	.0000000		.9000000+000
BLOCK 6	.1053605-02	.0000000		.9000000+000
BLOCK 7	.1053605-02	.0000000		.9000000+000
BLOCK 8	.1053605-02	.0000000		.9000000+000
RELIABILITY OF THE SYSTEM THRU TIME				100. HOURS = .800

Fig. 11. Page 2 of output for example 1

PROBABILITY TREE 8						
8	5	4	2	1		
8	5	4	-2	3	1	
8	5	-4	3	1		
8	5	-4	-3	6	2	1
8	-5	7	3	1		
8	-5	7	-3	4	2	1
8	-5	7	-3	-4	6	2 1
8	-5	-7	6	2	1	

Fig. 12. Probability tree

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	77	78	79	80	
0	1																																							
0	2																																							
0	3	0	2																																					
0	4																																							
0	5	0	4																																					
0	6	0	1	0	3																																			
0	7	0	6	0	5																																			
1	0	0	7																																					9
0	2																																							
0	6	0	2	0	3																																			
0	7	0	4	0	5																																			
0	0																																							
	1	+	1	.	0																																			
0	1	.	0	1																																				
0	2	.	0	0	5																																			
	:																																							
	:																																							
1	0	0	.	0																																				
	.	0	1																																					
0	6																																							0
0	7																																							0
																																								7
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	77	78	79	80	

Fig. 13. Input for example 2

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	77	78	79	80
1	+	1	.	0																																			
0	1	.	0	1																																			
1	0	0	.	0																																			
0	6																																						1
0	7																																						1
																																							7
	1	+	1	.	0																																		
0	1	.	0	1																																			
1	0	0	.	0																																			
0	6		.	0	0	0	1																															2	
0	7		.	0	0	0	1																															2	
																																							7
	1	+	1	.	0																																		
0	1	.	0	1																																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	77			
1	+	1	.	0																																			
0	1	.	0	1																																			
1	0	0	.	0																																			
0	6		.	0	0	0	1																															2	
0	7		.	0	0	0	1																															2	
																																							7
	1	+	1	.	0																																		
0	1	.	0	1																																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	77			
1	+	1	.	0																																			
0	1	.	0	1																																			

Fig. 13 (contd)

Fig. 13 (contd)

```

RELIABILITY BLOCK DIAGRAM

BLOCK 1  INPUT
        OUTPUT 6

BLOCK 2  INPUT
        OUTPUT 3

BLOCK 3  INPUT 2
        OUTPUT 6

BLOCK 4  INPUT
        OUTPUT 5

BLOCK 5  INPUT 4
        OUTPUT 7

BLOCK 6  INPUT 1 3
        OUTPUT 7

BLOCK 7  INPUT 6 5
        OUTPUT 10

BLOCK 10 INPUT 7
        OUTPUT

SENSE SWITCH 6 CONTROLS THE FOLLOWING BLOCKS
2 3

SENSE SWITCH 7 CONTROLS THE FOLLOWING BLOCKS
4 5

```

Fig. 14. Page 1 of output for example 2

	ACTIVE F/R	DORMANT F/R	R-INITIAL	RELIABILITY
BLOCK 1	.1000000-01	.0000000		.3678794+000
BLOCK 2	.5000000-02	.5000000-04		
BLOCK 3	.5000000-02	.5000000-04		
BLOCK 4	.5000000-02	.5000000-04		
BLOCK 5	.5000000-02	.5000000-04		
BLOCK 10	.0000000	.0000000		.1000000+001
BLOCK 51				.5790860+000
BLOCK 52				.6912851+000
SENSE SWITCH	ACTIVE F/R	DORMANT F/R	PROBABILITY	
BLOCK 6	.0000000	.0000000	1.0000000	
BLOCK 7	.0000000	.0000000	1.0000000	
RELIABILITY OF THE SYSTEM THRU TIME			100. HOURS =	.9179

Fig. 15. Output for switch option 0 for example 2

	ACTIVE F/R	DORMANT F/R	R-INITIAL	RELIABILITY
BLOCK 1	.1000000-01	.0000000		.3678794+000
BLOCK 2	.5000000-02	.5000000-04		
BLOCK 3	.5000000-02	.5000000-04		
BLOCK 4	.5000000-02	.5000000-04		
BLOCK 5	.5000000-02	.5000000-04		
BLOCK 10	.0000000	.0000000		.1000000+001
BLOCK 51				.5211774+000
BLOCK 52				.6010675+000
SENSE SWITCH ACTIVE F/R DORMANT F/R PROBABILITY				
BLOCK 6	.0000000	.0000000	.9000000	
BLOCK 7	.0000000	.0000000	.9000000	
RELIABILITY OF THE SYSTEM THRU TIME			100. HOURS =	.879

Fig. 16. Output for switch option 1 for example 2

	ACTIVE F/R	DORMANT F/R	R-INITIAL	RELIABILITY
BLOCK 1	.1000000-01	.0000000		.3678794+000
BLOCK 2	.5000000-02	.5000000-04		
BLOCK 3	.5000000-02	.5000000-04		
BLOCK 4	.5000000-02	.5000000-04		
BLOCK 5	.5000000-02	.5000000-04		
BLOCK 10	.0000000	.0000000		.1000000+001
BLOCK 51				.5762049+000
BLOCK 52				.6865014+000
SENSE SWITCH ACTIVE F/R DORMANT F/R PROBABILITY				
BLOCK 6	.0000000	.1000000-03	1.0000000	
BLOCK 7	.0000000	.1000000-03	1.0000000	
RELIABILITY OF THE SYSTEM THRU TIME			100. HOURS =	.9160

Fig. 17. Output for switch option 2 for example 2

	ACTIVE F/R	DORMANT F/R	R-INITIAL	RELIABILITY
BLOCK 1	.1000000-01	.0000000		.3678794+000
BLOCK 2	.5000000-02	.5000000-04		
BLOCK 3	.5000000-02	.5000000-04		
BLOCK 4	.5000000-02	.5000000-04		
BLOCK 5	.5000000-02	.5000000-04		
BLOCK 10	.0000000	.0000000		.1000000+001
BLOCK 51				.3636460+000
BLOCK 52				.4795264+000
SENSE SWITCH ACTIVE F/R DORMANT F/R PROBABILITY				
BLOCK 6	.1000000-01	.1000000-03	1.0000000	
BLOCK 7	.1000000-01	.1000000-03	1.0000000	
RELIABILITY OF THE SYSTEM THRU TIME			100. HOURS =	.791

Fig. 18. Output for switch option 3 for example 2

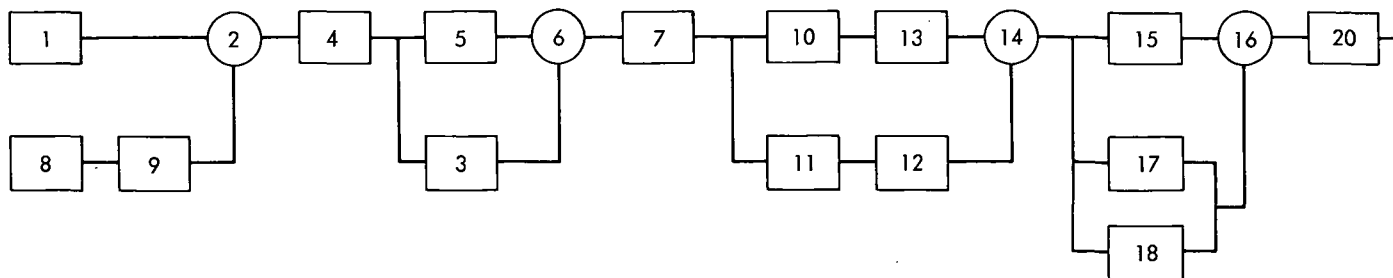


Fig. 19. Reliability block diagram with standby redundancy

RELIABILITY BLOCK DIAGRAM			
BLOCK 1	INPUT		
	OUTPUT	2	
BLOCK 2	INPUT	1	9
	OUTPUT	4	
BLOCK 3	INPUT	4	
	OUTPUT	6	
BLOCK 4	INPUT	2	
	OUTPUT	3	5
BLOCK 5	INPUT	4	
	OUTPUT	6	
BLOCK 6	INPUT	5	3
	OUTPUT	7	
BLOCK 7	INPUT	6	
	OUTPUT	10	11
BLOCK 8	INPUT		
	OUTPUT	9	
BLOCK 9	INPUT	8	
	OUTPUT	2	
BLOCK 10	INPUT	7	
	OUTPUT	13	
BLOCK 11	INPUT	7	
	OUTPUT	12	
BLOCK 12	INPUT	11	
	OUTPUT	14	
BLOCK 13	INPUT	10	
	OUTPUT	14	
BLOCK 14	INPUT	13	12
	OUTPUT	15	17 18
BLOCK 15	INPUT	14	
	OUTPUT	16	
BLOCK 16	INPUT	15	17 18
	OUTPUT	20	
BLOCK 17	INPUT	14	
	OUTPUT	16	
BLOCK 18	INPUT	14	
	OUTPUT	16	
BLOCK 20	INPUT	16	
	OUTPUT		
SENSE SWITCH 2 CONTROLS THE FOLLOWING BLOCKS			
		8	9
SENSE SWITCH 6 CONTROLS THE FOLLOWING BLOCKS			
		3	
SENSE SWITCH 14 CONTROLS THE FOLLOWING BLOCKS			
		11	12
SENSE SWITCH 16 CONTROLS THE FOLLOWING BLOCKS			
		17	18

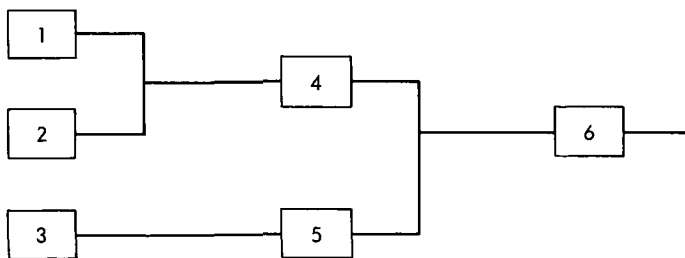
Fig. 20. Output for example 3

	ACTIVE F/R	DORMANT F/R	R-INITIAL	RELIABILITY
BLOCK 1	.5000000-02	.0000000		.6065307+000
BLOCK 3	.5000000-02	.5000000-03		.7499374+000
BLOCK 4	.5000000-02	.0000000		.6065307+000
BLOCK 5	.5000000-02	.0000000		.6065307+000
BLOCK 7	.5000000-02	.0000000		.6065307+000
BLOCK 8	.5000000-02	.5000000-03		
BLOCK 9	.5000000-02	.5000000-03		
BLOCK 10	.5000000-02	.0000000		.6065307+000
BLOCK 11	.5000000-02	.5000000-03		
BLOCK 12	.5000000-02	.5000000-03		
BLOCK 13	.5000000-02	.0000000		.6065307+000
BLOCK 15	.5000000-02	.0000000		.6065307+000
BLOCK 17	.5000000-02	.5000000-03		
BLOCK 18	.5000000-02	.5000000-03		
BLOCK 20	.5000000-02	.0000000		.6065307+000
BLOCK 51				.5741910+000
BLOCK 52				.5524726+000
BLOCK 53				.9274769+000
SENSE SWITCH	ACTIVE F/R	DORMANT F/R	PROBABILITY	
BLOCK 2	.0000000	.2000000-04	1.0000000	
BLOCK 6	.0000000	.5000000-04	1.0000000	
BLOCK 14	.0000000	.5000000-04	1.0000000	
BLOCK 16	.0000000	.3000000-04	1.0000000	
RELIABILITY OF THE SYSTEM THRU TIME				100. HOURS = .1166675

Fig. 20 (contd)

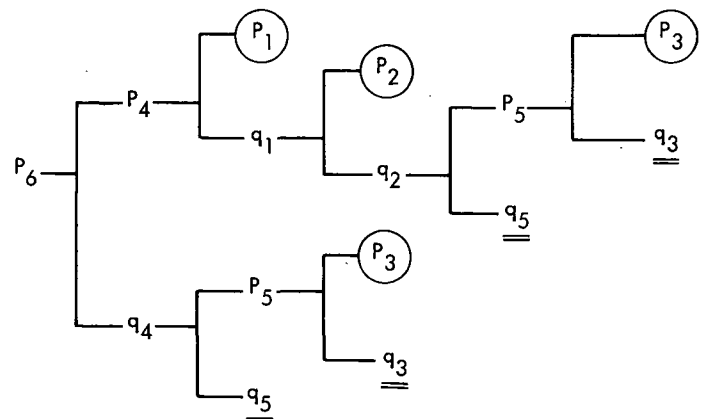
B. Theory

1. **Active redundancy.** The probability of successful operation for a system involving active redundancy can be found using the "probability tree" method (Ref. 2). Many examples of this method can be found in Ref. 1, where it was used to compute reliability, so we will use only one example here. Consider the diagram below.



The probability tree method for finding the system reliability equation for the preceding diagram is to begin at the output or success block (number 6) and work toward the inputs, searching out success paths. For example, the first path searched out is $p_6 p_4 p_1$, which represents the probability that blocks 1, 4, and 6 are successful. If block 1 is not successful, with probability q_1 , then we must search

out a new path, which is $p_6 p_4 q_1 p_2$. If block 2 also fails, then $p_6 p_4 q_1 q_2 p_5 p_3$ is a success path. This listing of the success paths is usually denoted in the form of a tree (hence a "probability tree"), as follows:



A circle indicates that we reached a success state via this path, while a double underline indicates that we reached a system failure; i.e., from this state there is no way to find a success state.

The probability of a particular success path occurring is simply the product of the probabilities in that path. The system reliability is then the sum of the probabilities for each of the success paths. Thus

$$P(\text{system}) = p_6 p_4 p_1 + p_6 p_4 q_1 p_2 \\ + p_6 p_4 q_1 q_2 p_5 p_3 + p_6 q_4 p_5 p_3$$

for this case.

The computation of active redundant configurations forms the basis for the whole program. The computation is handled in two phases. First, the system equation is derived in the subroutine TREE. Second, the numeric computations with this equation are performed in the subroutine SYSP.

Subroutine SYSP is straightforward, but a few remarks concerning the method for the equation derivation subroutine may be in order. The TREE subroutine is based on the probability tree method described earlier. The objective of the subroutine is to derive the probability tree and store it in ISAVE (I, J), where $J = 1, \dots$ holds the entries of the I th success path, and $I = 1, \dots$ denotes the success paths. Failure paths in the tree are not saved since they are not needed for computation.

The input for TREE is the block diagram contained in IB (I, J, NB), where

IB(1, 1, NB) = number of inputs to NB

IB(1, 2, NB) = number of outputs of NB

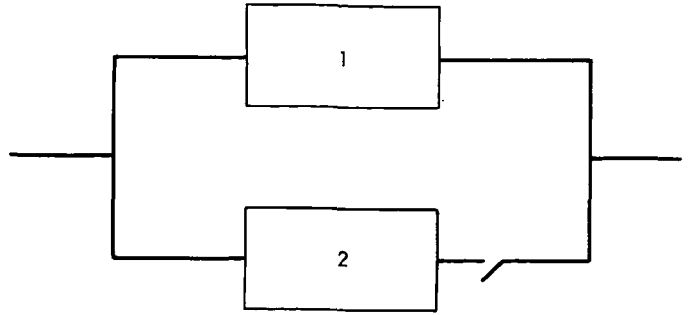
IB(I, 1, NB), $I = 2, \dots$ = the block numbers of the inputs to NB

IB(I, 2, NB), $I = 2, \dots$ = the block numbers of the outputs of NB

2. Standby redundancy. The principle that is used in computing standby redundancy is very simple. However, difficulty occurs in applying the principle to complex circuits.

We now treat some simple cases to motivate the use of what we call \bar{R} . The probability that a set of blocks will

work if needed is termed \bar{R} . Consider the simplest standby circuit:



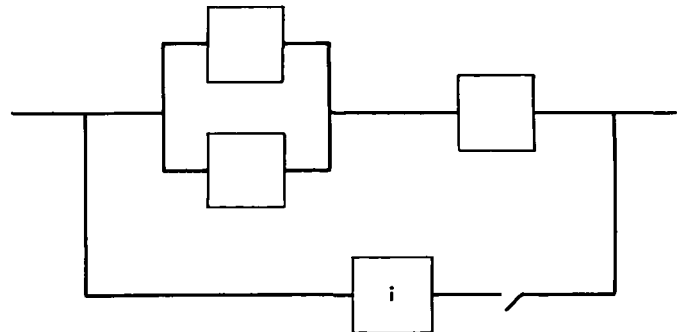
Here, system probability

$$P_{\text{sys}} = P(1) + P(1 \text{ fails and } 2 \text{ works}) = P(1) + P(1^c \cdot 2) \\ = P(1) + P(1^c) \cdot P(2 | 1^c) \quad (1)$$

where $P(1)$ is the probability that block 1 works, 1^c is 1 complement, and $P(1^c)$ is the probability that block 1 fails. For the time-independent case (active redundancy), this equation reduces to $P(1) + [1 - P(1)] \cdot P(2)$. But in the standby case this is not so. We define $\bar{R}_2 = P(2 | 1^c)$ so that Eq. (1) becomes $P(1) + P(1^c) \cdot \bar{R}_2$. Note that \bar{R}_2 can also be written as

$$\bar{R}_2 = \frac{P(2 \cap 1^c)}{P(1^c)}$$

Now we extend the concept of \bar{R} to a more complicated circuit. Assume that a block i is in standby redundancy to a circuit (not necessarily only one block) as in the following diagram:



In this case assume the exponential distribution for each of the blocks. (The exponential is assumed here as an example since it is the most common; \bar{R} could be developed just as easily using any distribution.) Also assume that

λ_i is the failure rate of block i and $P_i(t)$ is the probability of needing block i at time t , i.e., the failure probability of the main path.

$$\bar{R}_i(t) = \frac{R_{i0} P_{i0} e^{-\lambda_i t} + R_{i0} \int_0^t e^{-\lambda_i(t-\tau)} \dot{P}_i(\tau) d\tau}{P_i(t)} \quad (2)$$

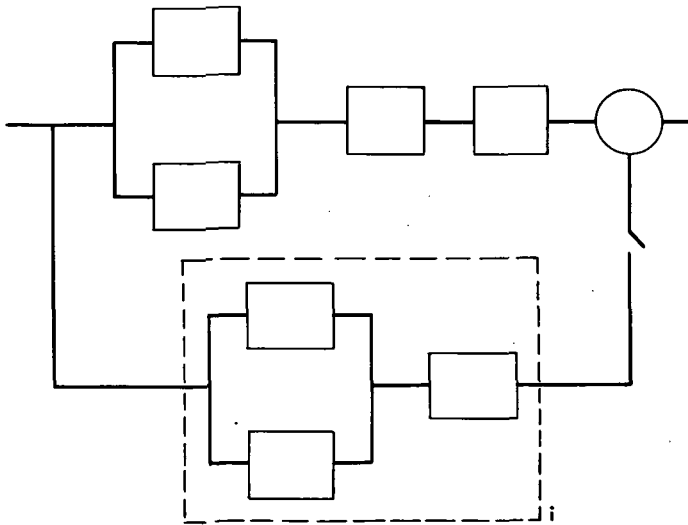
Note that the first term (with P_{i0}) takes care of the initial boundary condition in case the main circuit fails at turn-on ($t = 0$). The probability of block i turning on, i.e., the initial reliability, is R_{i0} , which can be set equal to 1 if this generalization is not used.

Note that \bar{R}_i can be rewritten as follows

$$\bar{R}_i(t) = \frac{R_{i0} P_{i0} e^{-\lambda_i t} + R_{i0} \int_0^{P_i(t)} e^{-\lambda_i(t-\tau)} dP_i(\tau)}{P_i(t)} \quad (3)$$

a form which is more suitable to computation.

It is possible to extend the concept of \bar{R} to cases where the standby portion of the circuit is not a single block but a diagram in itself. Consider the following:



Define the block diagram of the standby circuit as a pseudoblock i . Now no matter how complex the prime

path or the standby path (pseudoblock i) is, the following generalization of \bar{R}_i is true.

$$\bar{R}_i(t) = \frac{R_{i0} P_{i0} R_i(t) + R_{i0} \int_0^{P_i(t)} R_i(t-\tau) dP_i(\tau)}{P_i(t)} \quad (4)$$

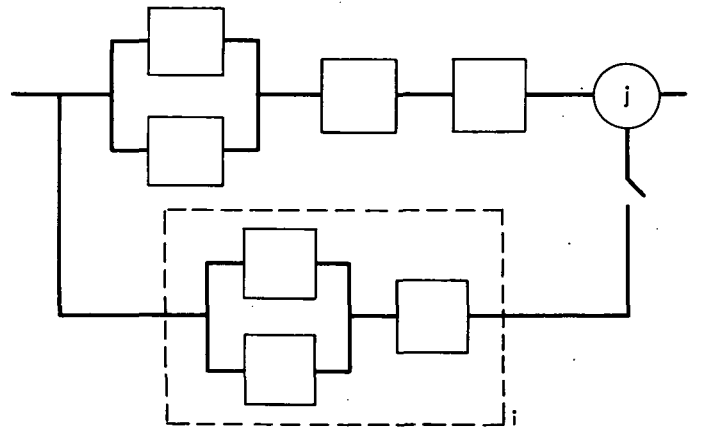
Note that R_i is the reliability of pseudoblock i . The equation for R_i can be derived using probability trees.

Equation (4) still does not consider the effects of dormancy and switching. The following equation extends \bar{R}_i to include dormancy.

$$\bar{R}_i(t) = \frac{R_{i0} P_{i0} R_i(t) + R_{i0} \int_0^{P_i(t)} R_{id}(\tau) R_i(t-\tau) dP(\tau)}{P_i(t)} \quad (5)$$

where $R_{id}(\tau)$ is the reliability of block i (or pseudoblock i) in the dormant state through time τ .

Switching must be considered in several steps because of the various switching options available. The following diagram is the same as the preceding diagram except that the sensing and switching block has been added.



Sense block j is not taken into account when the overall probability tree is computed. The block does not assume a serial role as the drawing might indicate; it is there only to indicate that standby redundancy is present. The sensing and switching are considered by direct inclusion of switch-

ing failure rates in the computation of \bar{R}_i of the standby system i . This is done as follows:

- (1) Switch option 0: Perfect switching. This is Eq. (5) directly, since the switching is assumed perfect.
- (2) Switch option 1: Constant probability that switch works. If we let P_{sj} be the probability that the sensing and switching of sense block j works, then our equation for R_i becomes

$$\bar{R}_i(t) = \frac{R_{i0} P_{i0} R_i(t) P_{sj} + R_{i0} P_{sj} \int_0^{P_i(t)} R_{id}(\tau) R_i(t - \tau) dP_i(\tau)}{P_i(t)} \quad (6)$$

- (3) Switch option 2: Dormant failure rate for switch. By this is meant that the sensing and switching hardware must survive until it is needed, i.e., until the time τ that the prime path fails and the standby system i is needed. Once the failure has been sensed and the standby system i switched in, it is assumed that the sensing and switching hardware j can fail, with no adverse effects on the system. For this case, the \bar{R}_i equation becomes

$$\bar{R}_i(t) = \frac{R_{i0} P_{i0} R_i(t) + R_{i0} \int_0^{P_i(t)} R_{id}(\tau) e^{-\lambda_{sdj}\tau} R_i(t - \tau) dP_i(\tau)}{P_i(t)} \quad (7)$$

where λ_{sdj} is the failure rate for the sensing and switching of block j for the period when the standby path is dormant.

- (4) Switch option 3: Dormant and active failure rate for switch. In addition to the requirement that the sensing and switching be required to last until needed if and when the prime units fail, this option also requires the sensing and switching to work the whole time that the standby path is active. This often occurs when the design is such that the switch mechanism requires power to hold the standby units switched in. The failure rate λ_s is the failure rate for the sensing and switching when the standby unit is active. ($\lambda_s = 0$ reverts back to switching option 2 if $\lambda_{sd} \neq 0$.) It is recognized that λ_s would often be specified different from λ_{sd} , and so it is required that both be entered. This requirement could arise for several reasons. For example, the switch hardware might be dormant when the

standby is dormant and active when the standby is active. Another reason could be that until the standby is needed (at time τ^1), both the sensing and switching are required; while after the standby is switched in as active, only the switch portion of the hardware might be needed and the failure-sensing portion might no longer be required. Thus it is required, with this option, to specify both λ_{sj} and λ_{sdj} . The equation for \bar{R}_i is

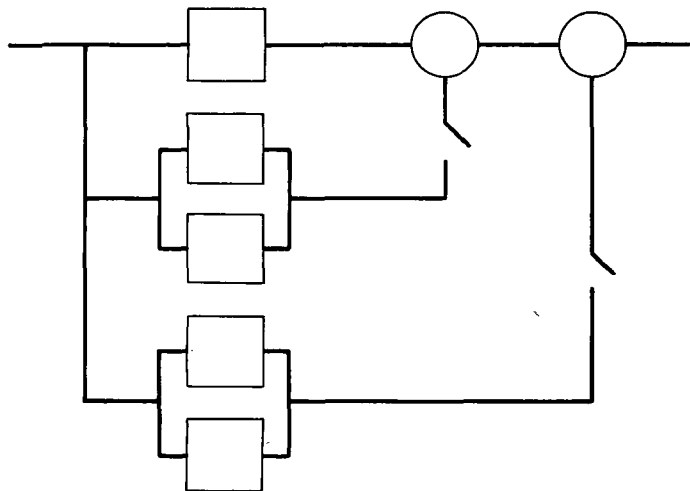
$$\bar{R}_i(t) = \frac{R_{i0} P_{i0} R_i(t) + R_{i0} \int_0^{P_i(t)} R_{id}(\tau) e^{-\lambda_{sdj}\tau} R_i(t - \tau) e^{-\lambda_{sj}(t-\tau)} dP_i(\tau)}{P_i(t)} \quad (8)$$

The importance of \bar{R} is that, when the probability tree method is used, one needs the probability that the redundant units will work if they are needed. In active redundant circuits, since the redundant blocks are turned on anyway, this probability is straightforward. For example, in Eq. (1), $P(2 | 1^c) = e^{-\lambda_2 t}$. When the redundant paths are in standby, the time of failure of the prime unit is important and the needed probability is not so straightforward. However, the probability needed in the probability tree development is simply \bar{R} . Thus when \bar{R} is used for standby blocks instead of the straightforward exponentials used for the active redundant blocks, the probability tree approach is still valid.

From a computation point of view, we proceed as follows to develop the system reliability equation. We develop a probability tree for the RBD by ignoring the fact that some of the blocks are in standby redundancy; that is, we consider all redundancy as active. In all probabilities in the probability tree that contain standby blocks, replace the associated p 's and q 's with \bar{R} and $(1 - \bar{R})$, respectively.

The problem is to derive \bar{R} for these standby blocks. Briefly, this is how it is accomplished. As was previously demonstrated, the new things needed for \bar{R}_i are P_i , the probability that standby block i is needed, and R_i , the reliability of block (or pseudoblock) i as a function of time. Consider the derivation of P_i , which is the probability of failure of the sub-RBD to which block i is standby. This can be found by generating the probability tree for this sub-RBD as follows. Generate a probability tree back from the sense block. Generate a probability tree back from the standby path. Subtract the probability tree of the standby path and the standby blocks out of the sense block probability tree. This leaves a probability tree for the sub-RBD only. Note that this works even when you have "stacked" standby redundancy (i.e., standby redundancy

parallel to a circuit with standby redundancy). This is shown as follows:



In such a case, when the probability tree for P_i of the outside standby is generated, \bar{R} will occur instead of p for the inside standby. From this probability tree, P_i can be computed as a function of time.

To find R_i , generate a probability tree for the standby path. This is done by generating a probability tree from the sense block, eliminating all entries not in the standby path. This will leave many duplicate entries in the probability tree, and duplicate probability tree paths are eliminated. Only active redundancy occurs in the standby path, so R_i can be easily computed as a function of time from this probability tree.

With R_i and P_i , \bar{R} can be generated. The probability P_i is not needed since in the numerical integration for \bar{R} this becomes ΔP_i , and P_i is known as a function of time.

3. Partial redundancy. Currently partial redundancy is handled by manually setting up the problem in terms of equivalent blocks as described earlier. Thus, from a

computation viewpoint, partial redundancy is an application of the equivalent block feature.

4. Equivalent blocks. Equivalent blocks occur when the same piece of physical hardware appears several times in the reliability block diagram. When such a situation occurs, the blocks are listed with different block numbers in order to avoid ambiguity when the RBD is described. Equivalent blocks are then designated by listing those blocks that are the same piece of hardware. This information is stored in the ITEM array in the program. When the system equation is computed, it is computed in terms of success paths. In a success path, it is possible to have two or more of a set of equivalent blocks. When this occurs, it indicates that if the block worked in one occurrence, it will work in the other and vice versa. If, for example, P_2, \dots, P_5 are both in a success path and 2 and 5 are equivalent, then $p(5 | 2 \text{ worked}) = 1$, and $p(5 | 2 \text{ did not work}) = 0$. With equivalent blocks present, these conditional probabilities are used when the system equation is computed.

C. Summary

The probability tree method, as developed in Ref. 1, provides a powerful tool for computing system reliability. It can handle complex systems that would ordinarily require tedious hand calculations, and it can handle nonstandard systems involving active redundancy inside standby redundancy. This is largely because the method *develops* the system reliability equation for any RBD presented – it does not simply combine built-in series and parallel reliability equations.

This report has extended the method of Ref. 1 to systems using standby redundancy. Because of the generality of the method, no restrictions need be placed on the types of active redundancy used inside either the prime or the standby path of a standby redundant system. The program listing for the computer program described can be found in Ref. 3.

References

1. Chelson, P., *Reliability Math Modeling Using the Digital Computer*, Technical Report 32-1089. Jet Propulsion Laboratory, Pasadena, Calif., Apr. 15, 1967.
2. *Reliability Engineering*, ARINC Research Corporation, edited by W. H. Von Alven, Prentice-Hall, Inc., 1965, pp. 279-282.
3. Chelson, Paul O., and Eckstein, R. Eric, *Program Listing for the reliability Block Diagram Computation Program of JPL Technical Report 32-1543*, Technical Memorandum 33-513. Jet Propulsion Laboratory, Pasadena, Calif., Dec. 15, 1971.